

In the Specification:

Please replace the paragraph beginning on page 1, line 16, with the following rewritten paragraph:

--There have been a large number of attempts to operate the same program on the platforms of different processing systems. A programming language JavaJAVA<sup>®</sup> (Java JAVA<sup>®</sup> is a registered trademark of Sun Microsystems Incorporated) is one of the answers. JavaJAVA<sup>®</sup> programming language realizes a platform-independent system, and excels in portability among different types of platforms.--

Please replace the paragraph beginning on page 1, line 24, with the following rewritten paragraph:

--The source code of a program expressed in JavaJAVA<sup>®</sup> programming language is normally processed in a syntax analysis, etc., converted into a binary file called byte code, and then distributed. A JavaJAVA<sup>®</sup> programming language byte code is a general-purpose instruction code independent of a platform of a processing system, interpreted and executed in a JavaJAVA<sup>®</sup> programming language executing system referred to as a ~~Java-VM~~JAVA<sup>®</sup> VM (virtual machine). When various processing systems are provided with the environment corresponding to the ~~Java-VM~~JAVA<sup>®</sup> VM, the same JavaJAVA<sup>®</sup> programming language byte code can be executed in different processing systems.--

Please replace the paragraph beginning on page 2, line 10, with the following rewritten paragraph:

--Described below is the technology of executing the JavaJAVA<sup>®</sup> programming language byte code in a processing system.--

Please replace the paragraph beginning on page 2, line 12, with the following rewritten paragraph:

--A JavaJAVA<sup>®</sup> programming language interpreter sequentially interprets a JavaJAVA<sup>®</sup> programming language byte code for each instruction, and allows a processing system to perform a process corresponding to the instruction. The JavaJAVA<sup>®</sup> programming language interpreter has merit of a program expressed by the JavaJAVA<sup>®</sup> programming language byte code, interpreted as is and processed, but has demerit of the time required to interpret an instruction, thereby slowing the process.--

Please replace the paragraph beginning on page 2, line 20, with the following rewritten paragraph:

--A method of compiling in advance the JavaJAVA<sup>®</sup> programming language byte code for a native code which is a machine language directly executable by a processing system can be used as a method for improving throughput of the program expressed in the JavaJAVA<sup>®</sup> programming language byte code, and a tool for the compiling process is

referred to as a native compiler. The throughput of a program can be considerably enhanced by compiling the JavaJAVA<sup>®</sup> programming language byte code into native code, but there is the problem that the portability among different platforms, which is the noticeable merit of Java JAVA<sup>®</sup> programming language, is lost. Furthermore, each time the JavaJAVA<sup>®</sup> programming language byte code is updated by changing a program, an operator of a processing system has to instruct the system to compile the JavaJAVA<sup>®</sup> programming language byte code into a native code.--

Please replace the paragraph beginning on page 3, line 10, with the following rewritten paragraph:

--A just-in-time compiler (hereinafter referred to as a 'JIT compiler') is suggested to maintain the merit of JavaJAVA<sup>®</sup> programming language, that is, the portability among different platforms, and improve the problem of the JavaJAVA<sup>®</sup> programming language interpreter in throughput.--

Please replace the paragraph beginning on page 3, line 15, with the following rewritten paragraph:

--When the JavaJAVA<sup>®</sup> programming language byte code is executed, the JIT compiler allows a processing system to sequentially execute the code while compiling the code into native code in a function (method in JavaJAVA<sup>®</sup> programming language) unit used

in the JavaJAVA<sup>®</sup> programming language byte code. At this time, a compiled native code is stored in the main memory. When a function already compiled into native code appears as the JavaJAVA<sup>®</sup> programming language byte code is compiled, the function is not recompiled, but the native code stored in the main memory is directly executed by the processing system, thereby shortening the processing time required by the compiling process. Normally, a function decelerating the process of the entire program is repeatedly called in many cases. Therefore, the throughput can be improved in this method using the JavaJAVA<sup>®</sup> programming language byte code.--

Please replace the paragraph beginning on page 4, line 6, with the following rewritten paragraph:

--Since the JIT compiler also optimizes an instruction expressed by the JavaJAVA<sup>®</sup> programming language byte code, which is difficult in an interpreting process performed by an interpreter, the optimization improves the throughput of the system. Furthermore, the JIT compiler is free of the complicated operations necessarily performed by an operator of the native compiler on the processing system.--

Please replace the paragraph beginning on page 4, line 14, with the following rewritten paragraph:

--A JIT compiler used for executing the JavaJAVA<sup>®</sup> programming language

byte code is popularly used, but it also can be configured such that a program expressed in another programming language and a program (generally referred to as a 'source program' in this specification) expressed in an intermediate language obtained by performing a syntax analysis and an optimizing process on a program can be compiled into the native code which can be directly interpreted by a specific platform, and executed on the platform.--

Please replace the paragraph beginning on page 4, line 25, with the following rewritten paragraph:

--As described above, the problem of the JIT compiler about the throughput can be solved while maintaining the merit of JavaJAVA<sup>®</sup> programming language, that is, the portability among different platforms. Furthermore, it has the merit that the load of operations to be performed by an operator is smaller than the load in the native code. However, since it is always necessary to compile native code when the execution of a source program starts, there has been the problem in the execution of a source program using the JIT compiler that there occurs a time lag until the process described in the source program is actually started.--

Please replace the paragraph beginning on page 5, line 18, with the following rewritten paragraph:

--One of the embodiments of the present invention is based on an apparatus for

compiling a source program into a machine language directly executable on a platform of a specific processing system, and executing the machine language using a just-in-time-compiler system. In the apparatus, the machine language obtained by compiling the source program is stored in a storage unit for ~~maintain~~maintaining the stored data for each function expressed in the source program although the supply voltage has dropped. Then, it is determined whether or not the machine language obtained by compiling the function described in the source program is stored in the storage unit. Based on the determination result, either the machine language obtained by compiling the source program or the machine language stored in the storage unit is directly executed on the platform of a specific processing system.--

Please replace the paragraph beginning on page 16, line 20, with the following rewritten paragraph:

--In the apparatus according to the above mentioned first or second aspect of the present invention, the source program 1 can be expressed by the JavaJAVA<sup>®</sup>programming language byte code. In this case, the function described in the source program 1 corresponds to the method by JavaJAVA<sup>®</sup>programming language.--

Please replace the paragraph beginning on page 17, line 1, with the following rewritten paragraph:

--The embodiments of the embodiment are described below by referring to the attached drawings. An example of embodying the present invention in a program execution apparatus for compiling and executing the JavaJAVA<sup>®</sup> programming language byte code is described below.--

Please replace the paragraph beginning on page 26, line 25, with the following rewritten paragraph:

--The entire configuration of the second example of the program execution device embodying the present invention is the same as that shown in FIG. 2, but in the components shown in FIG. 2, the data stored in the hard disk device 26 are different from those of the first example. FIG. 6 shows the data stored in the hard disk device in the second example of the program execution device. The hard disk device 26 stores the compiled storage module 43 as in the first example, and also stores in advance a compiled standard module 51 obtained by compiling the method definition which can be described in the program source module 41. As the method definition which is compiled into a native code and stored in the hard disk device 26 as the compiled standard module 51, the method definition which is certified in an optional JavaJAVA<sup>®</sup> programming language execution system, such as the method definition which belongs to the class contained in the javaJAVA<sup>®</sup> programming language package which is a JavaJAVA<sup>®</sup> programming language standard class library, can be applicable.--

Please replace the paragraph beginning on page 29, line 10, with the following rewritten paragraph:

--Described below is the figure shown in FIG. 9. FIG. 9 is a figure showing the  
third example of the program execution device embodying the present invention.--